Sparse Approximation by Semidefinite Programming

Ali Çivril

February 10, 2017

Abstract

The problem of sparse approximation and the closely related compressed sensing have received tremendous attention in the past decade. Primarily studied from the viewpoint of applied harmonic analysis and signal processing, there have been two dominant algorithmic approaches to this problem: Greedy methods called the matching pursuit (MP) and the linear programming based approaches called the basis pursuit (BP). The aim of the current paper is to bring a fresh perspective to sparse approximation by treating it as a combinatorial optimization problem and providing an algorithm based on the powerful optimization technique semidefinite programming (SDP). In particular, we show that there is a randomized algorithm based on a semidefinite relaxation of the problem with performance guarantees depending on the coherence and the restricted isometry constant of the dictionary used. We then show a derandomization of the algorithm based on the method of conditional probabilities.

1 Introduction

Linear systems are encountered frequently in engineering and mathematical sciences. The fact that these systems are ill-conditioned or underdetermined in most applications led researchers to come up with regularizing constraints. One of the most useful and trending approaches is to require sparsity. In this approach, one seeks an approximate solution to a linear system while requiring that the unknown vector has few nonzero entries: Given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with $m < n$ (so-called a redundant dictionary) with unit norm columns, recover a sparse vector $\boldsymbol{b} \in \mathbb{R}^m$ by solving

$$\text{minimize } \|\boldsymbol{x}\|_0 \qquad \text{subject to } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

In this form of the problem, one is interested in finding the unknown vector $\boldsymbol{x}$, trying to recover $\boldsymbol{b}$. Another formulation of the problem, which is also of our main interest in this paper is

$$\text{minimize } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \qquad \text{subject to } \|\boldsymbol{x}\|_0 = k.$$

which we call the *sparse approximation* problem. Here, one is interested in minimizing a certain objective function representing the quality of the solution given a fixed sparsity level.

Stated in linear algebraic terms, the sparse approximation problem is about picking a k-dimensional subspace defined by k column vectors of $\boldsymbol{A}$ such that the orthogonal projection of $\boldsymbol{b}$ onto that subspace is as close as possible to $\boldsymbol{b}$. The problem can be defined in full generality using notions from functional analysis (e.g. Hilbert spaces with elements representing functions), as is usually conceived in signal processing. Indeed, defined in Hilbert and Banach spaces, it has been studied as *highly nonlinear approximation* in functional approximation theory [24, 25]. However, we consider the finite dimensional case in this paper for which the linear algebraic language will be sufficient.

Sparse approximation problem is of combinatorial nature in finite dimensions and the optimal solution can be found by checking all $\binom{n}{k}$ subspaces. It is natural to ask whether one can do significantly better and the answer is most likely to be no since it is $\mathcal{NP}$-hard under general dictionaries [8, 20].

Most algorithmic approaches developed so far work on both variants of the problem mentioned above. However, the main focus seems to be on recovering the unknown vector $\boldsymbol{x}$ exactly under certain conditions. Since the problem is $\mathcal{NP}$-hard, one of the main algorithmic approaches to the problem of sparse recovery is to solve a convex relaxation of the problem:

$$\text{minimize } \|\boldsymbol{x}\|_1 \qquad \text{subject to } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

via linear programming. These type of approaches are called *basis pursuit* in the literature. Early results [5, 9, 11] established the fact that if the columns of $\boldsymbol{A}$ satisfy certain bounds on *coherence* (a term that will be defined shortly), the linear program above successfully recovers $\boldsymbol{x}$. Relatively more recent results [3, 4] provided similar results under the *restricted isometry property* (RIP).

The other main venue of attack is the greedy type algorithms which are variants of *matching pursuit* (MP) [19]. By far, the most popular one from which many other variants have been developed is the *orthogonal matching pursuit* (OMP) [8]. Contemporary variants of such pursuit methods include StOMP [12], rOMP [22], CoSAMP [21], subspace pursuit (SP) [6], gOMP [29], MMP [15] and POMP [23]. These works establish exact recovery conditions under coherence or RIP parameters, or they provide bounds on the distance between the actual vector $\boldsymbol{x}$ and the recovered vector $\tilde{\boldsymbol{x}}$.

Recent work considering the problem introduced new perspectives for solving it exactly via mixed-integer programming [2]. A heuristic method based on evolutionary computing has also been proposed [16], which is an example on the growing interest on the problem from different fields. For a much more comprehensive treatment of various approaches developed for sparse approximation, we refer the reader to the survey paper by Tropp and Wright [28].

In order to put the performance bounds that we will present in this paper into perspective, it is convenient to classify the types of analytical results related to the problem. There are roughly four types of algorithmic results regarding the sparse approximation problem as listed below.

1. Results relating the quality of the solution $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ produced by the algorithm to the quality of the optimal solution $\|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2$ given that $\boldsymbol{A}$ has bounded coherence and restricted isometry constant (e.g. [13]).

2. Results from functional approximation theory, showing the rate of convergence of an algorithm for elements from a specific set related to the dictionary (e.g. [17]).

3. Results stating conditions under which an algorithm optimally recovers a signal either via norms of certain matrices related to the dictionary (e.g. [27]) or the RIP of the dictionary (e.g. [7]).

4. Results bounding the norm of the distance between the actual unknown vector and the vector recovered by the algorithm (e.g. [21]).

The results of this paper is related to the first kind, which can be regarded as an *approximation algorithm* for the metric $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$. The use of the term approximation algorithm here is not arbitrary. This is the common terminology in the field of theoretical computer science and algorithm analysis, which is used for finding provably close solutions to the optimum where the exact solution is out of reach, e.g. when we have $\mathcal{NP}$-hardness. The results of Type I are also called Lebesgue-type

inequalities [10]. Before formally defining this notion, we first give the definitions of coherence and restricted isometry property which will be useful in the rest of the paper.

The *coherence* $\boldsymbol{\mu}$ of a dictionary $\boldsymbol{A}$ is defined as

$$\boldsymbol{\mu}(\boldsymbol{A}) = \max_{i \neq j} |\langle \boldsymbol{A}_i, \boldsymbol{A}_j \rangle|$$

where $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$ are the ith and jth columns of $\boldsymbol{A}$, respectively and $\langle \cdot, \cdot \rangle$ denotes the usual inner product defined on $\mathbb{R}^M$. Recall that the columns of a dictionary have unit norm. Hence, the coherence μ takes values in the $[0, 1]$ closed interval. Note that if the coherence is 0, we have an orthonormal dictionary, which is the most desirable case.

A dictionary $\boldsymbol{A}$ is said to satisfy RIP of order k if there exists a constant $\delta(\boldsymbol{A}) \in (0, 1)$ such that

$$(1 - \delta(\boldsymbol{A}))\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{A}\boldsymbol{x}\|_2^2 \leq (1 + \delta(\boldsymbol{A}))\|\boldsymbol{x}\|_2^2$$

for any $\boldsymbol{x}$ satisfying $\|\boldsymbol{x}\|_0 = k$. The minimum of all constants $\delta(\boldsymbol{A})$ satisfying these inequalities is called the restricted isometry constant $\delta_k(\boldsymbol{A})$ of $\boldsymbol{A}$. If $\delta_k(\boldsymbol{A}) = 0$, we have that $\boldsymbol{A}$ is orthonormal. In a sense, coherence and RIP measures how far a dictionary is from orthonormality.

To discuss the previous approximation algorithms for the sparse approximation problem, we define the following:

Definition 1. *An algorithm is an $(f(k), g(k))$-approximation algorithm for sparse approximation under coherence $h(k)$ if it selects a vector $\boldsymbol{x}$ with at most $g(k)$ nonzero elements from the dictionary $\boldsymbol{A}$ with $\mu(\boldsymbol{A}) \leq h(k)$ such that*

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \leq f(k) \cdot \|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2$$

where $\boldsymbol{x}^*$ is an optimal solution with at most k nonzero elements.

OMP is a well studied greedy algorithm yielding such approximation guarantees. There is also a slight variant of this algorithm named *orthogonal least squares* (OLS). In the last decade, the following results were found in a series of papers by different authors:

Theorem 2. *[13] OMP is an $(8\sqrt{k}, k)$-approximation algorithm for sparse approximation under coherence $\frac{1}{8\sqrt{2}(k+1)}$.*

Theorem 3. *[27] OMP is a $(\sqrt{1 + 6k}, k)$-approximation algorithm for sparse approximation under coherence $\frac{1}{3k}$.*

Theorem 4. *[10] OMP is a $(24, \lfloor k \log k \rfloor)$-approximation algorithm for sparse approximation under coherence $\frac{1}{90k^{3/2}}$.*

Theorem 5. *[26] OMP is a $(3, 2^{\lfloor \frac{1}{\delta} \rfloor} k)$-approximation algorithm for sparse approximation under coherence $\frac{1}{14\left(2^{\lfloor \frac{1}{\delta} \rfloor} k\right)^{1+\delta}}$, for any fixed $\delta > 0$.*

Theorem 6. *[18] OLS is a $(3, 2k)$-approximation algorithm for sparse approximation under coherence $\frac{1}{20k}$.*

In this paper, we present approximation algorithms for the sparse approximation problem with respect to the *differential objective function* by introducing a new algorithmic approach, namely semidefinite programming (SDP). More specifically, instead of trying to minimize the distance of

$\boldsymbol{b}$ to $\boldsymbol{Ax}$, we aim to maximize the orthogonal projection of $\boldsymbol{b}$ onto the space spanned by $\boldsymbol{Ax}$. Let $\text{supp}(\boldsymbol{x})$ denote the support of the vector $\boldsymbol{x}$, which is the set of its nonzero indices. Let $\boldsymbol{A_x}$ be the column sub-matrix of $\boldsymbol{A}$ with indices from $\text{supp}(\boldsymbol{x})$. Our goal is to solve the problem

$$\text{maximize } \|\boldsymbol{A_x A_x^+ b}\|_2 \qquad \text{subject to } \|\boldsymbol{x}\|_0 = k.$$

where $\boldsymbol{A_x^+}$ is the Moore-Penrose generalized pseudo-inverse of $\boldsymbol{A_x}$. To this aim, we define

Definition 7. *An algorithm is an $(f(k), g(k))$-differential approximation algorithm for the sparse approximation problem under coherence $h(k)$ (resp. under restricted isometry constant $h(k)$) if it selects a vector $\boldsymbol{x}$ with at most $g(k)$ nonzero elements from the dictionary $\boldsymbol{A}$ with $\mu(\boldsymbol{A}) \leq h(k)$ (resp. $\delta_k(\boldsymbol{A}) \leq h(k)$) such that*

$$\|\boldsymbol{A_x A_x^+ b}\|_2 \geq f(k) \cdot \|\boldsymbol{A_{x^*} A_{x^*}^+ b}\|_2$$

where $\boldsymbol{A_x}^*$ is the solution determined by an optimal $\boldsymbol{x}^*$ with at most k nonzero elements.

In the next section, we will provide an SDP based algorithm, which gives bounds according to this definition. The approximation ratio will depend on either the coherence or the restricted isometry constant of the dictionary (but not on k).

2 Approximation by SDP

Semidefinite programming is a powerful tool in combinatorial optimization, which is a generalization of linear programming. A semidefinite relaxation of a combinatorial optimization problem can also be written as a *vector program* where the number of vectors is equal to their dimensions and the objective function together with the constraints are linear in the dot products of these vectors. In order to formulate the problem where we try to optimize the differential objective function, we design a matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times n}$ which "selects" k column vectors from $\boldsymbol{A}$ such that

$$\boldsymbol{A_x} = \boldsymbol{AY}.$$

We could have selected $\boldsymbol{Y}$ to be of size $n \times k$ so that $\boldsymbol{AY}$ is of size $m \times k$, which correctly represents the column sub-matrix of interest. However, for the purposes of presenting an SDP relaxation, $\boldsymbol{Y}$ should be a square matrix as noted above. In this case, $\boldsymbol{AY}$ contains k columns of $\boldsymbol{A}$ and all other entries are 0. Assume without loss of generality that these reside in the first k columns of $\boldsymbol{AY}$. Then, $(\boldsymbol{AY})^+$ has the first k *rows* filled and others 0. Recall also that for a full-rank sub-matrix $\boldsymbol{C}$, we have $\boldsymbol{C}^+ = (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T$ so that if

$$\boldsymbol{AY} = \left(\begin{array}{c|c} \boldsymbol{C} & \boldsymbol{0} \end{array}\right)$$

Then

$$(\boldsymbol{AY})^+ = \left(\frac{\boldsymbol{C}^+}{\boldsymbol{0}}\right)$$

Thus, the square of the objective function becomes

$$\begin{aligned}
\|\boldsymbol{A_x A_x^+ b}\|_2^2 &= \left((\boldsymbol{AY})(\boldsymbol{AY})^+\boldsymbol{b}\right)^T \left((\boldsymbol{AY})(\boldsymbol{AY})^+\boldsymbol{b}\right) \\
&= \boldsymbol{b}^T \left((\boldsymbol{AY})(\boldsymbol{AY})^+\right)^T (\boldsymbol{AY})(\boldsymbol{AY})^+\boldsymbol{b} \\
&= \boldsymbol{b}^T \left((\boldsymbol{AY})\left((\boldsymbol{AY})^T(\boldsymbol{AY})\right)^{-1}(\boldsymbol{AY})^T\right)^T (\boldsymbol{AY})\left((\boldsymbol{AY})^T(\boldsymbol{AY})\right)^{-1}(\boldsymbol{AY})^T\boldsymbol{b} \qquad (1) \\
&= \boldsymbol{b}^T(\boldsymbol{AY})\left((\boldsymbol{AY})^T(\boldsymbol{AY})\right)^{-1}(\boldsymbol{AY})^T(\boldsymbol{AY})\left((\boldsymbol{AY})^T(\boldsymbol{AY})\right)^{-1}(\boldsymbol{AY})^T\boldsymbol{b} \\
&= \boldsymbol{b}^T(\boldsymbol{AY})\left((\boldsymbol{AY})^T(\boldsymbol{AY})\right)^{-1}(\boldsymbol{AY})^T\boldsymbol{b}.
\end{aligned}$$

Note here that the disturbing term is $\left((\boldsymbol{AY})^T(\boldsymbol{AY})\right)^{-1}$. If the columns of $\boldsymbol{A}$ were all orthogonal, this matrix would be equal to

$$\left(\begin{array}{c|c} \boldsymbol{I}_k & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} \end{array}\right)$$

leaving us with the term

$$\boldsymbol{b}^T(\boldsymbol{AY})(\boldsymbol{AY})^T\boldsymbol{b} = (\boldsymbol{b}^T\boldsymbol{A})\boldsymbol{Y}\boldsymbol{Y}^T(\boldsymbol{A}^T\boldsymbol{b}) \tag{2}$$

whose square root will be called the *simplified objective function*. We will see that the problem with this objective function can be solved via an SDP. Since the original objective function has an extra term which we are not be able to incorporate into the semidefinite formulation of the problem, it is necessary to relate these two quantities.

Lemma 8. *Let the restricted isometry constant of $\boldsymbol{A}$ be $\delta_k(\boldsymbol{A}) = \delta_k$. Then, the optimal value of the simplified objective function is not less than $\sqrt{1-\delta_k}$ times the optimal value of the original objective function.*

Proof. By the definition of the restricted isometry constant, the singular values of $\boldsymbol{AY}$ lie in the interval $\left[\sqrt{1-\delta_k}, \sqrt{1+\delta_k}\right]$. Thus, the eigenvalues of $(\boldsymbol{AY})^T(\boldsymbol{AY})$ lie in the interval $[1-\delta_k, 1+\delta_k]$. Since this matrix is symmetric, the same holds for its singular values. Note that the square of the simplified objective function is the dot product of a vector with itself. In the square of the original objective function, the aforementioned vector is multiplied by the matrix $\left((\boldsymbol{AY})^T(\boldsymbol{AY})\right)^{-1}$ whose singular values are in the range $\left[\frac{1}{1+\delta_k}, \frac{1}{1-\delta_k}\right]$, which scales the square of the simplified objective function by a factor of at most $\frac{1}{1-\delta_k}$. □

Lemma 9. *Let the coherence of $\boldsymbol{A}$ be $\mu(\boldsymbol{A}) = \frac{c}{k-1}$ for some $0 \leq c < 1$. Then, the optimal value of the simplified objective function is not less than $\sqrt{1-c}$ times the optimal value of the original objective function.*

Proof. Note that $(\boldsymbol{AY})^T(\boldsymbol{AY})$ contains the pairwise dot products of k columns in the upper left corner. The diagonals of this $k \times k$ sub-matrix are all 1. Thus, by the Gershgorin Disc Theorem [14], the eigenvalues of $(\boldsymbol{AY})^T(\boldsymbol{AY})$ lie in the interval $\left[1-(k-1)\cdot\frac{c}{k-1}, 1+(k-1)\cdot\frac{c}{k-1}\right] = [1-c, 1+c]$. Since this matrix is symmetric, the same bounds hold for its singular values, too. Consequently, the singular values of $\left((\boldsymbol{AY})^T(\boldsymbol{AY})\right)^{-1}$ are in the range $\left[\frac{1}{1+c}, \frac{1}{1-c}\right]$, which scales the square of the simplified objective function by a factor of at most $\frac{1}{1-c}$. □

Having these lemmas, we shall concentrate on the simplified objective function. Assign $\boldsymbol{d} = \boldsymbol{A}^T\boldsymbol{b}$. Then, our objective function in (2) becomes $\boldsymbol{d}^T\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{d}$. Our problem can then be recast as the following quadratic integer program:

$$\begin{aligned} \text{maximize} \quad & \boldsymbol{d}^T\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{d} && \text{(QIP)} \\ \text{subject to} \quad & \sum_{i=1}^{n} \boldsymbol{Y}_i^T\boldsymbol{Y}_i = k, \\ & \boldsymbol{Y}_i^T\boldsymbol{Y}_j = 0 \qquad \text{for } i \neq j. \end{aligned}$$

where $\boldsymbol{Y}_i$s are the *rows* of $\boldsymbol{Y}$ and each of them contains at most one 1 with other entries being 0. Note that this models our problem since $\boldsymbol{Y}$ should contain exactly k 1s. We also have that the

pairwise dot products of these rows are all 0. We relax this QIP to an SDP by allowing $\boldsymbol{Y}_i$s to take arbitrary vector values. The objective function is linear in the dot products of $\boldsymbol{Y}_i$s and further

$$\boldsymbol{d}^T \boldsymbol{Y}\boldsymbol{Y}^T \boldsymbol{d} = \sum_{i=1}^{n}\sum_{j=1}^{n}(d_i d_j)\boldsymbol{Y}_i^T \boldsymbol{Y}_j = \sum_{i=1}^{n} d_i^2 \boldsymbol{Y}_i^T \boldsymbol{Y}_i,$$

since $\boldsymbol{Y}_i^T \boldsymbol{Y}_j = 0$ for $i \neq j$. The choice of the matrix $\boldsymbol{Y}$ should also be clear by now since in an SDP we have n vectors of dimension n. In particular, their dimension cannot be $k < n$. Our SDP relaxation is given as:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} d_i^2 \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle && \text{(SDP-1)} \\ \text{subject to} \quad & \sum_{i=1}^{n} \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle = k, \\ & \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0 \qquad \text{for } i \neq j. \end{aligned}$$

By using standard SDP solvers, we can solve this relaxation in polynomial time. However, the main hurdle against a correct algorithmic result is that the vectors returned by the solver do not necessarily form a permutation matrix with 0s and 1s. As an extreme case, all $\boldsymbol{Y}_i$ might have norm $\sqrt{k/n}$ and they might still be orthogonal to each other, thus satisfying the constraints of the SDP. For the correct algorithmic solution, we want to select k rows of $\boldsymbol{Y}$. Consider the vectors returned by the solver and let $a_i = \boldsymbol{Y}_i^T \boldsymbol{Y}_i$. Then, the objective function is simply

$$\sum_{i=1}^{n} a_i d_i^2.$$

We propose an algorithm based on the idea of random sampling with judicious choice of probabilities. The algorithm considers a_is as probabilities (Note that $0 \leq a_i \leq 1$). For each i, the algorithm selects $\boldsymbol{Y}_i$ with probability a_i independently at random, and makes one of its entries 1, all others 0. It performs this operation for all rows independently by making sure that there is at most one 1 in each column of $\boldsymbol{Y}$.

Algorithm 1: SDP based randomized algorithm for sparse approximation

Let $\boldsymbol{Y} \in \mathbb{R}^{n \times n}$ be a matrix of all 0s.
Solve (SDP-1) to get the values $a_1, \ldots, a_n$.
for $i = 1$ *to* n **do**
  Select $\boldsymbol{Y}_i$ with probability a_i independently at random
end
Order the selected rows as $\boldsymbol{Y}_{i_1}, \ldots, \boldsymbol{Y}_{i_\ell}$ such that $i_1 \leq \ldots \leq i_\ell$.
for $j = 1$ *to* ℓ **do**
  Let the jth entry of $\boldsymbol{Y}_{i_j}$ be 1.
end
return $\boldsymbol{AY}$

The *expected* number of rows selected by the algorithm is clearly

$$\sum_{i=1}^{n} \mathbf{Pr}[\boldsymbol{Y}_i \text{ is selected}] = \sum_{i=1}^{n} a_i = k.$$

And the expected cost is

$$\mathbb{E}[W] = \sum_{i=1}^{n} \mathbf{Pr}[\boldsymbol{Y}_i \text{ is selected}] \cdot d_i^2 = \sum_{i=1}^{n} a_i d_i^2 = OPT,$$

where W is the random variable for the the cost of the solution returned by the algorithm and OPT is the cost of an optimal solution.

3 Derandomization

The disadvantage of the algorithm described in the previous section is that the cost of the solution and more importantly, the number of column vectors selected depends on the randomization introduced by the algorithm. We derandomize this algorithm via the standard method of conditional probabilities at the expense of solving several SDPs. In particular, we show that one can guarantee the same results with probability 1 by selecting exactly k column vectors.

In the method of conditional probabilities, we decide the inclusion of each column vector one by one making sure that the solution has cost at least OPT. We will proceed in steps and arrive at the result after n steps. We first show how the method works in the first step, and then in the ith step for some $2 \leq i \leq n$. The end result will be easy to see by simple induction. Recall that W is the random variable for the total cost of the solution picked. Then, we have

$$\mathbb{E}[W] = \mathbf{Pr}[a_1 = 1] \cdot \mathbb{E}[W|a_1 = 1] + \mathbf{Pr}[a_1 = 0] \cdot \mathbb{E}[W|a_1 = 0].$$

Suppose that we can compute $\mathbb{E}[W|a_1 = 1]$ and $\mathbb{E}[W|a_1 = 0]$. Take the larger one by breaking the tie arbitrarily. Suppose it is $\mathbb{E}[W|a_1 = 1]$. Then, we have

$$\begin{aligned}
&\mathbb{E}[W|a_1 = 1] \\
&\geq \mathbf{Pr}[a_1 = 1] \cdot \mathbb{E}[W|a_1 = 1] + \mathbf{Pr}[a_1 = 0] \cdot \mathbb{E}[W|a_1 = 0] \\
&= \mathbb{E}[W]
\end{aligned}$$

since $\mathbf{Pr}[a_1 = 1] + \mathbf{Pr}[a_1 = 0] = 1$ and $\mathbb{E}[W|a_1 = 1] \geq \mathbb{E}[W|a_1 = 0]$. But, the aforementioned conditional expectations can be computed by solving appropriately formulated SDPs. In particular, $\mathbb{E}[W|a_1 = 1]$ can be found by solving

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} d_i^2 \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle && \text{(SDP-2)} \\
\text{subject to} \quad & \langle \boldsymbol{v}_1, \boldsymbol{v}_1 \rangle = 1, \\
& \sum_{i=2}^{n} \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle = k - 1, \\
& \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0 \qquad \text{for } i \neq j.
\end{aligned}$$

Similarly, $\mathbb{E}[W|a_1 = 0]$ can be found by solving

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} d_i^2 \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle && \text{(SDP-3)} \\
\text{subject to} \quad & \langle \boldsymbol{v}_1, \boldsymbol{v}_1 \rangle = 0, \\
& \sum_{i=2}^{n} \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle = k, \\
& \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0 \qquad \text{for } i \neq j.
\end{aligned}$$

Suppose, we have selected k_i column vectors by the beginning of the ith step. Then, we have

$$\begin{aligned}
&\mathbb{E}[W|a_1 = b_1, \ldots, a_{i-1} = b_{i-1}] \\
&= \mathbf{Pr}[a_i = 1] \cdot \mathbb{E}[W|a_1 = b_1, \ldots, a_{i-1} = b_{i-1}, a_i = 1] \\
&+ \mathbf{Pr}[a_i = 0] \cdot \mathbb{E}[W|a_1 = b_1, \ldots, a_{i-1} = b_{i-1}, a_i = 0].
\end{aligned}$$

where $b_j = 0$ or $b_j = 1$ for $1 \leq j \leq i-1$ and $\sum_{j=1}^{i-1} b_j = k_i$. If we can compute the conditional probabilities above and select the larger one, say the first, we have

$$\begin{aligned}
&\mathbb{E}[W|a_1 = b_1, \ldots, a_{i-1} = b_{i-1}, a_i = 1] \\
&\geq \mathbf{Pr}[a_i = 1] \cdot \mathbb{E}[W|a_1 = b_1, \ldots, a_{i-1} = b_{i-1}, a_i = 1] \\
&+ \mathbf{Pr}[a_i = 0] \cdot \mathbb{E}[W|a_1 = b_1, \ldots, a_{i-1} = b_{i-1}, a_i = 0] \\
&= \mathbb{E}[W|a_1 = b_1, \ldots, a_{i-1} = b_{i-1}]
\end{aligned}$$

by similar reasoning to the first step. The conditional probability $\mathbb{E}[W|a_1 = b_1, \ldots, a_{i-1} = b_{i-1}, a_i = 1]$ can be computed by solving the SDP

$$\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} d_j^2 \langle \boldsymbol{v}_j, \boldsymbol{v}_j \rangle && \text{(SDP-4)} \\
\text{subject to} \quad & \langle \boldsymbol{v}_j, \boldsymbol{v}_j \rangle = b_j && \text{for } 1 \leq j \leq i-1, \\
& \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle = 1, \\
& \sum_{j=i+1}^{n} \langle \boldsymbol{v}_j, \boldsymbol{v}_j \rangle = k - k_i - 1, \\
& \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0 && \text{for } i \neq j.
\end{aligned}$$

And $\mathbb{E}[W|a_1 = b_1, \ldots, a_{i-1} = b_{i-1}, a_i = 0]$ can be computed by solving the SDP

$$\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} d_j^2 \langle \boldsymbol{v}_j, \boldsymbol{v}_j \rangle && \text{(SDP-5)} \\
\text{subject to} \quad & \langle \boldsymbol{v}_j, \boldsymbol{v}_j \rangle = b_j && \text{for } 1 \leq j \leq i-1, \\
& \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle = 0, \\
& \sum_{j=i+1}^{n} \langle \boldsymbol{v}_j, \boldsymbol{v}_j \rangle = k - k_i, \\
& \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0 && \text{for } i \neq j.
\end{aligned}$$

We continue performing these operations until we select all the k column vectors after n steps. Note that, we might have to solve $2n$ SDPs in total. It is easy to see that by simple induction, we have

$$\mathbb{E}[W|a_1 = b_1, \ldots, a_n = b_n] \geq \mathbb{E}[W],$$

where $b_j = 0$ or $b_j = 1$ for $1 \leq j \leq n$ and $\sum_{j=1}^{n} b_j = k$. Thus, we have a deterministic algorithm selecting exactly k column vectors with cost at least $E[W] = OPT$. Combining these facts with Lemma 8 and Lemma 9, we have the following theorems:

Theorem 10. *Let the restricted isometry constant of $\boldsymbol{A}$ be $\delta_k(\boldsymbol{A}) = \delta_k$. Then, there is an SDP based algorithm for sparse approximation, which selects k column vectors represented by $\boldsymbol{A_x}$ such that*

$$\|\boldsymbol{A_x}\boldsymbol{A_x}^{+}\boldsymbol{b}\|_2 \geq \sqrt{1-\delta_k} \cdot \|\boldsymbol{A_{x^*}}\boldsymbol{A_{x^*}}^{+}\boldsymbol{b}\|_2,$$

where $\boldsymbol{A_x}^*$ is the solution determined by an optimal $\boldsymbol{x^*}$ with at most k nonzero elements. In other words, the algorithm is a $\left(\sqrt{1-\delta_k}, k\right)$-differential approximation algorithm for sparse approximation under restricted isometry constant δ_k.

Theorem 11. *Let the coherence of $\boldsymbol{A}$ be $\mu(\boldsymbol{A}) = \frac{c}{k-1}$ for some $0 \leq c < 1$. Then, there is an SDP based algorithm for sparse approximation, which selects k column vectors represented by $\boldsymbol{A_x}$ such that*

$$\|\boldsymbol{A_x}\boldsymbol{A_x}^{+}\boldsymbol{b}\|_2 \geq \sqrt{1-c} \cdot \|\boldsymbol{A_{x^*}}\boldsymbol{A_{x^*}}^{+}\boldsymbol{b}\|_2,$$

where $\boldsymbol{A_x}^*$ is the solution determined by an optimal $\boldsymbol{x^*}$ with at most k nonzero elements. In other words, the algorithm is a $\left(\sqrt{1-c}, k\right)$-differential approximation algorithm for sparse approximation under coherence $\frac{c}{k-1}$.

One can convert these results into approximations for the usual objective function.

Corollary 12. *Let the restricted isometry constant of $\boldsymbol{A}$ be $\delta_k(\boldsymbol{A}) = \delta_k$. Then, there is an SDP based algorithm for sparse approximation, which selects k column vectors (i.e. $\|\boldsymbol{x}\|_0 = k$) such that*

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \delta_k \cdot \|\boldsymbol{b}\|_2^2 + (1-\delta_k) \cdot \|\boldsymbol{A}\boldsymbol{x^*} - \boldsymbol{b}\|_2^2,$$

where $\boldsymbol{x^*}$ is an optimal solution with at most k nonzero elements.

Proof. By the Pythagoras Theorem, we have

$$\|\boldsymbol{A_x}\boldsymbol{A_x}^{+}\boldsymbol{b}\|_2^2 = \|\boldsymbol{b}\|_2^2 - \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

And, similarly

$$\|\boldsymbol{A_{x^*}}\boldsymbol{A_{x^*}}^{+}\boldsymbol{b}\|_2^2 = \|\boldsymbol{b}\|_2^2 - \|\boldsymbol{A}\boldsymbol{x^*} - \boldsymbol{b}\|_2^2.$$

By Theorem 10, we have

$$\|\boldsymbol{A_x}\boldsymbol{A_x}^{+}\boldsymbol{b}\|_2^2 \geq (1-\delta_k) \cdot \|\boldsymbol{A_{x^*}}\boldsymbol{A_{x^*}}^{+}\boldsymbol{b}\|_2^2.$$

Combining all these expressions and rearranging suitably yields the desired result. □

A similar result with the same proof holds for the coherence:

Corollary 13. *Let the coherence of $\boldsymbol{A}$ be $\mu(\boldsymbol{A}) = \frac{c}{k-1}$ for some $0 \leq c < 1$. Then, there is an SDP based algorithm for sparse approximation, which selects k column vectors (i.e. $\|\boldsymbol{x}\|_0 = k$) such that*

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq c \cdot \|\boldsymbol{b}\|_2^2 + (1-c) \cdot \|\boldsymbol{A}\boldsymbol{x^*} - \boldsymbol{b}\|_2^2,$$

where $\boldsymbol{x^*}$ is an optimal solution with at most k nonzero elements.

3.1 Comparison of SDP with OMP

In this subsection, we would briefly like to discuss the SDP approach from a qualitative viewpoint, particularly with respect to the choice of column vectors as described in the derandomized algorithm. Let us first compare the result of the last corollary with one of the bounds proven for OMP in [27], namely

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq (1 + 6k) \cdot \|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2^2.$$

for coherence $\mu(\boldsymbol{A}) \leq \frac{1}{3k}$. For the same coherence, our result translates into

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \frac{1}{3} \cdot \|\boldsymbol{b}\|_2^2 + \frac{2}{3} \cdot \|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2^2.$$

Note that the results are not directly comparable as we need further information relating $\|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2$ to $\|\boldsymbol{b}\|_2$. However, we know that for the correct sparsity level k, $\|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2$ will be 0 or very close to 0. In this case, the performance guarantee of the SDP based algorithm is much worse than that of OMP. If on the other hand, k is much smaller than the correct sparsity level (one can think of the sparsity level being large here), SDP guarantees a better solution provided that

$$\|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2^2 \geq \frac{\|\boldsymbol{b}\|_2^2}{1 + 18k}.$$

It has been frequently discussed in the literature that the weakness of OMP is mainly due to the "wrong" choices of column vectors at the beginning. Indeed, many contemporary pursuit based methods have been devised to circumvent this drawback. In this respect, the SDP based approach provides, as discussed in the section describing the derandomization, a completely new way of selecting column vectors one by one, arguably in a much more subtle and sophisticated manner. It is thus an important conceptual issue to determine what advantages and disadvantages this approach possesses compared to the greedy methods. As roughly discussed above, the SDP based approach seems to be behaving better with respect to the selection of the first few column vectors.

4 Final Remarks

The main purpose of this paper was to make a conceptual contribution by showing a new algorithmic approach to sparse approximation. Naturally, we neither discussed efficiency issues nor presented experimental results. It would be interesting to see if there are fast implementations for solving the SDPs we mentioned, particularly by the recent primal-dual methods introduced by Arora and Kale [1]. We also suspect that there are more refined versions of our approach providing guarantees in other different forms. As noted in the previous subsection, comparison and possible combinations with greedy approaches is also an open venue of research.

References

[1] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. *J. ACM*, 63(2):12, 2016.

[2] S. Bourguignon, J. Ninin, H. Carfantan, and M. Mongeau. Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance. *IEEE T. Signal Proces.*, 64(6):1405–1419, 2016.

[3] E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pur. Appl. Math.*, 59:1207–1223, 2006.

[4] E. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE T. Inform. Theory*, 52(12):5406–5425, 2006.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, 2001.

[6] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing: Closing the gap between performance and complexity. *IEEE T. Inform. Theory*, 55(5):2230–2249, 2009.

[7] M. A. Davenport and M. B. Wakin. Analysis of orthogonal matching pursuit using the restricted isometry property. *IEEE T. Inform. Theory*, 56(9):4395–4401, 2010.

[8] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constr. Approx.*, 13:57–98, 1997.

[9] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *P. Nat. Acad. Sci. USA*, 100:2197–2202, 2003.

[10] D. L. Donoho, M. Elad, and V. N. Temlyakov. On lebesgue-type inequalities for greedy approximation. *J. Approx. Theory*, 147:185–795, 2007.

[11] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE T. Inform. Theory*, 47(7):2845–2862, 2001.

[12] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE T. Inform. Theory*, 58(2):1094–1121, 2012.

[13] A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss. Approximation of functions over redundant dictionaries using coherence. In *Proceedings of the 14th Annual ACM-SIAM Aymposium on Discrete Algorithms (SODA)*, pages 243–252, 2003.

[14] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge Univ. Press, 1985.

[15] S. Kwon, J. Wang, and B. Shim. Multipath matching pursuit. *IEEE T. Inform. Theory*, 60(5):2986–3001, 2014.

[16] L. Li, X. Yao, R. Stolkin, M. Gong, and S. He. An evolutionary multiobjective approach to sparse reconstruction. *IEEE T. Evolut. Comput.*, 18(6):827–845, 2014.

[17] E. Liu and V. N. Temlyakov. The orthogonal super greedy algorithm and applications in compressed sensing. *IEEE T. Inform. Theory*, 58(4):2040–2047, 2012.

[18] E. D. Livshitz. On the optimality of the orthogonal greedy algorithm for μ-coherent dictionaries. *J. Approx. Theory*, 164(5):668–681, 2012.

[19] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE T. Signal Proces.*, 41(12):3397–3415, 1993.

[20] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, 1995.

[21] D. Needell and J. A. Tropp. CoSAMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. A.*, 26(3):301–321, 2009.

[22] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Found. Comput. Math.*, 9(3):317–334, 2009.

[23] O. Teke, A. C. Gürbüz, and O. Arikan. Perturbed orthogonal matching pursuit. *IEEE T. Signal Proces.*, 61(24):6220–6231, 2013.

[24] V. N. Temlyakov. Greedy algorithms and M-term approximation with regard to redundant dictionaries. *J. Approx. Theory*, 98:117–145, 1999.

[25] V. N. Temlyakov. Weak greedy algorithms. *Adv. Comput. Math.*, pages 213–227, 2000.

[26] V. N. Temlyakov and Pavel Zheltov. On performance of greedy algorithms. *J. Approx. Theory*, 163(9):1134–1145, 2011.

[27] J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE T. Inform. Theory*, 50(10):2231–2242, 2004.

[28] J. A. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. *P. IEEE*, 98(6):948–958, 2010.

[29] J. Wang, S. Kwon, and B. Shim. Generalized orthogonal matching pursuit. *IEEE T. Signal Proces.*, 60(12):6202–6216, 2012.